08 | 测试数据：是不是可以把所有的参数都保存到Excel中?

2020-02-17 陈磊

讲述：陈磊
时长 09:35　大小 7.69M

你好，我是陈磊。

课程到现在，我们已经一起从接口测试思维的训练，走到了接口测试技术的训练，随着学习的不断深入，你应该也有了一个自己的测试框架，虽然这个框架可能还很简陋。但是任何事情不管多晚开始，都好于从未开始，因此学到现在，你已经迈出了接口测试以及其测试技术的第一步。

做任何事情，从零到一都需要莫大的勇气和坚定的决心，在这个过程中，你要将自己从舒适区，进入一个陌生的领域，这确实很难。但如果你和我一起走到了这一节课，那么我要恭喜你，你已经完成了接口测试从零到一的转变，后续从一到无穷大，你只需要随着时间去积累经验就可以了。

如果把接口测试比喻成要炒一盘菜的话，那么我在之前的全部课程中，重点都是在讲解如何完成接口测试，也就是教你如何炒菜。我也教过你如何解决接口测试的需求，为你提供了解决问题的能力和手段，这也就是在帮你建造一个设备齐全的厨房，帮你一起完成接口测试任务 。

有了精致的厨房后，我也告诉了你要怎么制作顶级的厨具，也就是接口测试的技术方法和实践方式。这些厨具既有锅碗瓢盆，也有刀勺铲叉，这里的锅碗瓢盆就是你的测试框架，刀勺铲叉就是你使用框架完成的测试脚本，这其中既包含了单接口的测试脚本，也包含了业务逻辑多接口测试脚本。

那么如果想炒菜你还需要准备什么呢？毫无疑问那就是菜，所谓“巧妇难为无米之炊”，即使你有高超的手艺，有世界顶级的厨具，但如果没有做菜的原材料，那也没办法把菜做出来，就算是世界顶级大厨，也无法完成这样的任务。

今天我就顺着这个思路，和你讲讲菜的准备，也就是接口测试的数据准备工作。

测试数据的好处：打造自动化测试框架

随着你不断封装自己的测试框架，你的框架就始终处于等米下锅这样一种的状态，而米就是测试数据。我在之前的课程中，都是将测试数据直接写在代码里，赋值给一个变量，然后通过接口测试逻辑来完成测试。

说到这，我还是把我们之前用过的”战场“这个系统拿出来，看一看它“选择武器”这个接口测试脚本（你可以回到04中查看），虽然你现在对怎么撰写、怎么封装类似的接口脚本都已经烂熟于心，但我们还是先看一下它的代码段：

复制代码

```
# uri_login存储战场的选择武器
uri_selectEq = '/selectEq'
# 武器编号变量存储用户名参数
equipmentid = '10003'
# 拼凑body的参数
payload = 'equipmentid=' + equipmentid
response_selectEq = comm.post(uri_selectEq,params=payload)
print('Response内容: ' + response_selectEq.text)
```

这就是你通过自己的 Common 类改造后的测试框架，但是现在，它还不能是算是一个完美的框架，为什么呢？

这是因为，你现在的参数都是直接通过 equipmentid 变量赋值的，在做测试的时候，你还需要不断修改这个参数的赋值，才能完成接口的入参测试，这不是一种自动化的测试思路。

因此，你需要将数据封装，通过一种更好的方式，将数据存储到一种数据存储文件中，这样代码就可以自行查找对应的参数，然后调取测试框架执行测试流程，接着再通过自动比对返回预期，检验测试结果是否正确。

这样做有两个好处。

1. 无人值守，节省时间和精力。我们将所有的参数都存储到外部存储文件中，测试框架就可以自行选择第一个参数进行测试，在完成第一个测试之后，它也就可以自行选择下一个参数，整个执行过程是不需要人参与的。否则的话，我们每复制一组参数，就要执行一次脚本，然后再人工替换一次参数，再执行一次脚本，这个过程耗时费力，而且又是一个纯人工控制的没什么技术含量的活动。
2. 自动检测返回值，提高测试效率。如果你用上面的代码段完成接口测试，就要每执行一次，人工去观察一次，看接口的返回是不是和预期一致，人工来做这些事情，不只非常耗费时间，效率也很低下。但是通过代码完成一些关键匹配却很容易，这可以大大提高测试效率，快速完成交付。

怎么样，看到这些好处，你是不是也想马上给你的框架加上数据处理的部分了呢？

如何选取测试数据

现在我们就马上开始动手，为你的框架加上参数类。

首先，你先要定义一种参数的存储格式。那么我想问你的是，要是让你选择把数据储存在一个文件中，你会选择什么格式的文件呢？

我相信你肯定和我的选择一样，用 Excel。因为目前来看，Excel 是在设计测试用例方面使用最多的一个工具，那么我们也就可以用 Excel 作为自己的参数存储文件。

但在动手之前，你也应该想到，你的参数文件类型不会是一成不变的 Excel，未来你也有可能使用其他格式的参数文件，因此在一开始你还要考虑到参数类的扩展性，这样你就不用每多了一种参数文件存储格式，就写一个参数类，来完成参数的选取和调用了。

那么如何选取和调用参数呢？你可以看看我设计的参数类：

复制代码

```
import json
import xlrd
class Param(object):
  def __init__(self,paramConf='{}'):
    self.paramConf = json.loads(paramConf)
  def paramRowsCount(self):
    pass
  def paramColsCount(self):
    pass
  def paramHeader(self):
    pass
  def paramAllline(self):
    pass
  def paramAlllineDict(self):
    pass


class XLS(Param):
  '''
  xls基本格式(如果要把xls中存储的数字按照文本读出来的话,纯数字前要加上英文单引号:

  第一行是参数的注释,就是每一行参数是什么

  第二行是参数名,参数名和对应模块的po页面的变量名一致

  第3~N行是参数

  最后一列是预期默认头Exp
  '''
  def __init__(self, paramConf):
    '''
    :param paramConf: xls 文件位置(绝对路径)
    '''
    self.paramConf = paramConf
    self.paramfile = self.paramConf['file']
    self.data = xlrd.open_workbook(self.paramfile)
    self.getParamSheet(self.paramConf['sheet'])
  def getParamSheet(self,nsheets):
    '''
    设定参数所处的sheet
    :param nsheets: 参数在第几个sheet中
```

```
    :return:
    '''
    self.paramsheet = self.data.sheets()[nsheets]
  def getOneline(self,nRow):
    '''
    返回一行数据
    :param nRow: 行数
    :return: 一行数据 []
    '''
    return self.paramsheet.row_values(nRow)
  def getOneCol(self,nCol):
    '''
    返回一列
    :param nCol: 列数
    :return: 一列数据 []
    '''
    return self.paramsheet.col_values(nCol)
  def paramRowsCount(self):
    '''
    获取参数文件行数
    :return: 参数行数 int
    '''
    return self.paramsheet.nrows
  def paramColsCount(self):
    '''
    获取参数文件列数(参数个数)
    :return: 参数文件列数(参数个数) int
    '''
    return self.paramsheet.ncols
  def paramHeader(self):
    '''
    获取参数名称
    :return: 参数名称[]
    '''
    return self.getOneline(1)
  def paramAlllineDict(self):
    '''
    获取全部参数
    :return: {{}},其中dict的key值是header的值
    '''
    nCountRows = self.paramRowsCount()
    nCountCols = self.paramColsCount()
    ParamAllListDict = {}
    iRowStep = 2
    iColStep = 0
    ParamHeader= self.paramHeader()
    while iRowStep < nCountRows:
    ParamOneLinelist=self.getOneline(iRowStep)
    ParamOnelineDict = {}
    while iColStep<nCountCols:
    ParamOnelineDict[ParamHeader[iColStep]]=ParamOneLinelist[iColStep]
    iColStep=iColStep+1
```

```
94      iColStep=0
95      ParamAllListDict[iRowStep-2]=ParamOnelineDict
96      iRowStep=iRowStep+1
97      return ParamAllListDict
98    def paramAllline(self):
99      '''
100     获取全部参数
101     :return: 全部参数[[]]
102     '''
103     nCountRows= self.paramRowsCount()
104     paramall = []
105     iRowStep =2
106     while iRowStep<nCountRows:
107     paramall.append(self.getOneline(iRowStep))
108     iRowStep=iRowStep+1
109     return paramall
110   def __getParamCell(self,numberRow,numberCol):
111     return self.paramsheet.cell_value(numberRow,numberCol)
112 class ParamFactory(object):
113   def chooseParam(self,type,paramConf):
114     map_ = {
115     'xls': XLS(paramConf)
116     }
117     return map_[type
```

上面这个代码看着很多，但你不需要完全看得懂，你只需要知道它解决问题的思路和方法就可以了，**思路就是通过统一抽象，建立一个公共处理数据的方式。**你可以设计和使用简单工厂类的设计模式，这样如果多一种参数存储类型，再添加一个对应的处理类就可以了，这很便于你做快速扩展，也可以一劳永逸地提供统一数据的处理模式。

如果你的技术栈和我不一样，那么你只需要搜索一下你自己技术栈所对应的简单工厂类设计模式，然后照猫画虎地把上面的逻辑实现一下就可以了。接下来，你就可以把这次测试的全部参数都存到 Excel 里面了，具体内容如下图所示：

用例	期望
equipmentid	exp
10001	your pick up equipmentid：10001
10002	your pick up equipmentid：10002
10003	your pick up equipmentid：10003

通过上面的参数类你可以看出，在这个 Excel 文件中，第一行是给人读取的每一列参数的注释，而所有的 Excel 都是从第二行开始读取的，第二行是参数名和固定的表示预期结果的 exp。现在，我们使用 ParamFactory 类，再配合上面的这个 Excel，就可以完成”战场“系统“选择武器”接口的改造了，如下面这段代码所示：

复制代码

```
#引入Common、ParamFactory类
from common import Common
from param import ParamFactory
import os
# uri_login存储战场的选择武器
uri_selectEq = '/selectEq'
comm = Common('http://127.0.0.1:12356',api_type='http')
# 武器编号变量存储武器编号，并且验证返回时是否有参数设计预期结果
# 获取当前路径绝对值
curPath = os.path.abspath('.')
# 定义存储参数的excel文件路径
searchparamfile = curPath+'/equipmentid_param.xls'
# 调用参数类完成参数读取，返回是一个字典，包含全部的excel数据除去excel的第一行表头说明
searchparam_dict = ParamFactory().chooseParam('xls',{'file':searchparamfile,'sl
i=0
while i<len(searchparam_dict):
  # 读取通过参数类获取的第i行的参数
  payload = 'equipmentid=' + searchparam_dict[i]['equipmentid']
  # 读取通过参数类获取的第i行的预期
  exp=searchparam_dict[i]['exp']
  # 进行接口测试
  response_selectEq = comm.post(uri_selectEq,params=payload)
  # 打印返回结果
  print('Response内容: ' + response_selectEq.text)
  # 读取下一行excel中的数据
  i=i+1
```

这样再执行你的测试脚本，你就可以看到数据文件中的三条数据，已经都会顺序的自动执行了。那么后续如果将它付诸于你自己的技术栈，以及自己的测试驱动框架比如 Python 的 unittest、Java 的 Junit等，你就可以通过断言完成预期结果的自动验证了。

总结

今天我们接口测试数据准备的内容就到这里了，在接口测试的工作中，作为“巧妇”的测试工程师，还是需要参数这个“米”来下锅的，虽然我们之前课程中的代码涉及到参数的处理，但是都很简单粗暴，一点也不适合自动化的处理方式，因此今天，我带你完成了参数类的封装。

有的时候，我们也把参数类叫做参数池，这也就是说参数是存放在一个池子中，那我们准备好的池子就是 Excel。我相信未来你也会不断扩展自己参数池的种类，这有可能是由于测试接口的特殊需求，也有可能是由于团队技术栈的要求。因此，我们封装参数池是通过简单工厂设计模式来实现的，如果你的代码基础并不好，那么你可以不用搞清楚简单工厂设计模式是什么，只需要知道如何模拟上述代码，再进行扩展就可以了。

一个好用的测试框架既要有很好的可用性，也要有很好的扩展性设计，这样我们的私有接口测试武器仓库就会变成可以不断扩展的、保持统一使用方法的武器仓库，这样才能让你或者你的团队在面对各种各样的测试任务时，既可以快速适应不同接口测试的需求，又不需要增加学习的成本。

思考题

今天我们一起学习了参数类的设计，并且将它应用到”战场“系统的接口测试脚本中，后续我又告诉你为了能够完成代码的自动验证，你需要引入一些测试驱动框架，那么，你的技术栈是什么？你在你的框架中选取的测试驱动框架又是什么呢？你能将之前”战场“系统的全流程测试脚本通过参数类完成改造吗？我期待看到你的测试脚本。

我是陈磊，欢迎你在留言区留言分享你的观点，如果这篇文章让你有新的启发，也欢迎你把文章分享给你的朋友，我们一起沟通探讨。

课程学习计划

关注极客时间服务号
每日学习签到

月领 25+ 极客币

【点击】保存图片，打开【微信】扫码>>>

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇　07 | WebSocket接口：如何测试一个完全陌生的协议接口？

下一篇　09 | 微服务接口：怎么用Mock解决混乱的调用关系？

精选留言 (16)

写留言

nothingc_lj
2020-02-18

接触了httprunner manager这个接口自动化框架平台感觉不错，是通过数据库来存储测试用例数据

> 作者回复: HttpRunner manager 和HttpRunner我曾经也学习过对应的代码，没有选择的原因其实我还是喜欢将框架的主动权掌握在自己手里面，不太喜欢follow定制化或者封装太强的平台。

3

孙瑜
2020-02-18

明天就按老师说的去实践下，过年宅家机缘开始学测试自动化，在现在公司刚一年目前测试大家都是辅以工具的纯手工测试，想提升一下团队测试效率和质量。调研后选了python进行学习，测试框架选了pytest，学了已有三周，目前根据公司业务已经把requests、selenium、ymal、db三方库做了封装组合，已开始实际进行api和web自动话测试。数据驱动目前只是用了一些pytest的参数化装饰器或者写sql去数据库随机抓一些id当参数，明...

展开

> 作者回复: 你的遇见了很多现实问题让我想起了我在经历京东那次技术转型。方法很多，第一，内部提升整体技术水平。第二，我看到您工时测试技术复杂度太高。确实需要降低一下，降低需要缩小你现在团队技术范围，selenium，po，api有点多。不要一起全推动；然后可以引入一下自动化写脚本的思考。这里面自动化写脚本的思路可以参考我之前再TICA[↑]的演讲内容，另一种提升内部技术水平，你可以看一下我们团队写的那本团队转型的书

2

null
2020-02-20

陈老师，建议例子打包一个项目，读者debug阅读代码比较深刻。从页面复制代码到本地运行，格式错了运行不起来。

展开

作者回复: 谢谢您的建议我这几天一起处理一下

1

hello world

2020-02-19

老师，请教个数据准备的问题，如果需要测试的是支付，那么是不是需要提前准备好订单数据，那订单数据本身可能又需要用户和商品数据，用户数据可能又需要地址数据，商品有可能需要品类和规格数据等等。这种情况下，是先构建最基础的数据比如地址数据并同时进行测试但是不删除数据一直留给支付测试使用，等支付使用完后再进行删除；还是每次测试支付的时候都使用一组选好的一直留在那里的数据进行测试呢。也就是如何构建...

展开

作者回复: 今后这种很靠后的数据，还是希望可以用其他接口测试脚本搞定。

1 1

、Attention

2020-02-17

老师你好，获取全部参数的方法是不是有点问题？？ 我一直执行不出来 。

作者回复: 您好我这边使用没遇见问题，您方便描述的具体一点吗。

5 1

ZeroIce

2020-02-17

个人觉得：Jenkins + postman不错

展开

作者回复: 不错不错，谢谢，是一个很好的组合。就是postman的脚步后期维护稍微有点吃力，不过也并不是说不可取。

1

罗春南_Nancy

2020-02-24

执行获取参数并访问选择装备的接口时，脚本是一直在执行状态的，但是没有返回任何结果

作者回复: 您方便看的battle系统有日志输出的控制台吗?

彦鋆
2020-02-23

老师，您好，看了您的文章，感觉茅塞顿开。我之前自己有写过前端的测试框架，利用po模式+selenium+unittest完成的；看了您的前几章，
我也初步写了自己的第一个接口自动化测试框架。利用自己的技术栈用分层处理的方式（有点借鉴PO模式）+unittest+excel存储数据实现了自己的接口自动化。
我的excel设计是这样的，每个sheet页的名称是接口的名称(比如battle中的index、logi...
展开

作者回复: 看了你详细的描述，已经很好了，可以将我运用的你的工作中了。架构设计上很好，继续努力，在工作中在逐渐的丰满框架的测试能力。

1

hancel
2020-02-21

def paramAllline(self):
 nCountRows= self.getCountRows() # 上面没有getCountRows()这个方法，看代码意思应该是paramRowsCount()

作者回复: 谢谢，这个代码是我引用错了，应该是paramRowsCount()

1

派森
2020-02-20

还是感觉yaml更加清晰，编写过程中格式问题也比较少
展开

作者回复: yaml，json，xml也是各有所长。但是选择合适自己的就好了。加油

ZYG
2020-02-19

老师，请问把测试数放在excel，是一个接口一个excel，还是放在同一个excel?

作者回复: 建议一个api一个excel

dd
2020-02-18

老师，在执行读取Excel测试用例的时候用的是循环调用一个方法去实现，在生成的测试报告中确并不是按用例去呈现测试结果，是需要对测试报告的封装代码做改动处理吗？能说说具体的实现思路吗？谢谢！

作者回复: 测试报告部分大概我们都不会自己写了，java会有maven的plugin完成，python也有很多，最简单的htmltestrumner，其实在测试过程中你会更加关注失败的case，因此是不是按照顺序执行我倒是觉得不是十几重要。如果你有这个需求可以尝试修改一下report部分代码。

小海
2020-02-18

你好！通过excel存储测试用例数据，每一行代表一个测试用例，这样循环读取测试用例使用一个测试方法取执行，测试报告中怎样区分每个用例的描述信息

展开

作者回复: 您好，测试报告中也可以通过数据驱动层的设计将其加入到你的测试报告中，就如同测试结果一样，但是这部分在这次课程中没有全部设计，处理方法和参数的处理方法一模一样，仅仅是使用地方不一样，因此没有特别将其讲出来。谢谢

aoe
2020-02-18

作为一个Java程序员，我希望自己的测试框架可以实现以下功能：
1. 发现Bug后立即报错，终止后续测试：快速失败机制，节省后续测试时间，可以立即进

去修改阶段；
2. 跑完全部测试用例，统计出所有Bug：适合多少提交后、自身粗心大意的程序员，及时知道增加的工作量，合理分配时间

展开

作者回复: 谢谢您，您的想法也给我了不同的思路。

happychap
2020-02-17

数据驱动最扎手的还是不幸遇到异构数据的情况，二维的Excel表和或关系数据库都没办法较好解决这种场景。数据项多的时候，维护测试数据也比较麻烦。可惜一直没能找到突破的法子，~ (ノ' - ')ノ ┻┻
老师有什么独门绝技分享一下嗄😊

展开

作者回复: 您说的确实是一个很难解决的问题，我曾经也遇见过，当初处理的方法就构建了一个数据生成服务，后续逐渐变成了我们team的测试数据服务也就是TDS，不知道你是不是可以借鉴一下。

2

VeryYoung
2020-02-17

在提取数据到excel过程中，像对数据库的检验数据应该怎么写比较合适？还有就是接口参数为空时的用例，这个用例有两种情况，一种是不传值另一种是什么都不传（name都不传）这个时候代码怎么根据excel来写？

作者回复: 什么都没有可以不在excel中写数据😀数据库校验不在参数部分，在代码逻辑的队数据库校验做封装。就是最原始的数据库查询封装😀可以找到您技术栈对应的类。